STATUS OF THE CLAIMS

1.   (Original) A method of executing a non-native software instruction, the method comprising:

receiving the non-native software instruction at a device;

generating a first native software instruction from a first instruction set based on the non-native software instruction, the generation of the first native software instruction occurring at the device;

executing the first native software instruction at the device;

generating a second native software instruction from a second instruction set based on the non-native software instruction, the generation of the second native software instruction occurring at the device; and

executing the second native software instruction at the device.

2.   (Original) A method as defined in claim 1, further comprising:

counting a number of times the first native software instruction is executed; and

comparing the number of times the first native software instruction is executed to a threshold, wherein generating the second native software instruction is in response to one of (i) the number equaling the threshold, and (ii) the number exceeding the threshold.

3.   (Original) A method as defined in claim 2, further comprising inserting instrumentation to count the number of times the first native software instruction is executed.

4.   (Original) A method as defined in claim 2, further comprising receiving the threshold via a mobile runtime configuration parameter.

5.   (Original) A method as defined in claim 1, wherein receiving the non-native software instruction at the device comprises receiving an intermediate language instruction at the device.

6.   (Original) A method as defined in claim 1, wherein receiving the non-native software instruction at the device comprises receiving Java byte code at the device.

7.   (Original) A method as defined in claim 1, wherein receiving the non-native software instruction at the device comprises wirelessly receiving the non-native software instruction at a hand-held computing device.

8.   (Original) A method as defined in claim 1, wherein the first native software instruction comprises an X-bit wide instruction, the second native software instruction comprises a Y-bit wide instruction, and X is less than Y.

9.   (Original) A method as defined in claim 1, wherein the first native software instruction comprises a 16-bit wide instruction, and the second native software instruction comprises a 32-bit wide instruction.

10.   (Original) A method as defined in claim 1, wherein the first native software instruction comprises a Thumb instruction, and the second native software instruction comprises an ARM instruction.

11.   (Original) A method as defined in claim 1, wherein generating the first native software instruction comprises compiling the non-native software instruction at the device using a just-in-time compiler.

12.   (Original) A method as defined in claim 1, further comprising:

configuring a first code optimization option prior to generation of the first native software instruction, the first code optimization option causing smaller code to be generated; and

configuring a second code optimization option prior to generation of the second native software instruction, the second code optimization option causing faster code to be generated.

13.   (Original) A method as defined in claim 1, wherein generating a first native software instruction comprises generating a first plurality of native software instructions, and generating a second native software instruction comprises generating a second plurality of native software instructions, the method further comprising:

counting a first number of instructions contained within the first plurality of native software instructions;

counting a second number of instructions contained within the second plurality of native software instructions; and

comparing the first number of instructions and the second number of instructions, wherein executing the first native software instruction is in response to one of (i) the second number of instructions equaling the first number of instructions and (ii) the second number of instructions exceeding the first number of instructions.

14.   (Original) A method as defined in claim 13, further comprising:

comparing the first number of instructions and the second number of instructions, wherein executing the second native software instruction is in response to the first number of instructions not exceeding the second number of instructions by more than a predetermined threshold.

15.   (Original) A method as defined in claim 1, further comprising:

measuring the first native software instruction resulting in a first number of bytes;

measuring the second native software instruction resulting in a second number of bytes; and

comparing the first number of bytes and the second number of bytes, wherein executing the first native software instruction is in response to the first number of bytes being less than the second number of bytes by at least a predetermined threshold.

16.   (Original) A method as defined in claim 1, further comprising:

measuring the first native software instruction resulting in a first number of bytes;

measuring the second native software instruction resulting in a second number of bytes; and

comparing the first number of bytes and the second number of bytes, wherein executing the second native software instruction is in response to the first number of bytes not being less than the second number of bytes by at least a predetermined threshold.

17. (Original) An article of manufacture comprising a machine-accessible medium having a plurality of machine accessible instructions that, when executed, cause a device to:

receive a non-native software instruction at the device;

generate a first native software instruction from a first instruction set based on the non-native software instruction, the generation of the first native software instruction occurring at the device;

execute the first native software instruction at the device;

generate a second native software instruction from a second instruction set based on the non-native software instruction, the generation of the second native software instruction occurring at the device; and

execute the second native software instruction at the device.

18. (Original) A machine-accessible medium as defined in claim 17, wherein the plurality of machine accessible instructions are structured to cause the device to:

count a number of times the first native software instruction is executed; and

compare the number of times the first native software instruction is executed to a threshold, wherein generating the second native software instruction is in response to one of (i) the number equaling the threshold, and (ii) the number exceeding the threshold.

19.   (Original) A machine-accessible medium as defined in claim 18, wherein the plurality of machine accessible instructions are structured to cause the device to insert instrumentation to count the number of times the first native software instruction is executed.

20.   (Original) A machine-accessible medium as defined in claim 17, wherein the non-native software instruction comprises an intermediate language instruction.

21.   (Original) A machine-accessible medium as defined in claim 17, wherein the first native software instruction comprises an X-bit wide instruction, the second native software instruction comprises a Y-bit wide instruction, and X is less than Y.

22.   (Original) A machine-accessible medium as defined in claim 17, wherein the first native software instruction comprises a Thumb instruction, and the second native software instruction comprises an ARM instruction.

23.   (Original) A machine-accessible medium as defined in claim 17, wherein the plurality of machine accessible instructions includes at least a portion of a just-in-time compiler.

24.   (Original) A machine-accessible medium as defined in claim 17, wherein the first native software instruction comprises a first plurality of native software instructions, and the second native software instruction comprises a second plurality of native software instructions, wherein the plurality of machine accessible instructions are structured to cause the device to:

count a first number of instructions contained within the first plurality of native software instructions;

count a second number of instructions contained within the second plurality of native software instructions; and

compare the first number of instructions and the second number of instructions, wherein executing the first native software instruction is in response to one of (i) the second number of instructions equaling the first number of instructions and (ii) the second number of instructions exceeding the first number of instructions.

25. (Original) A machine-accessible medium as defined in claim 24, wherein the plurality of machine accessible instructions are structured to cause the device to:

compare the first number of instructions and the second number of instructions, wherein executing the second native software instruction is in response to the first number of instructions exceeding the second number of instructions by more than a predetermined threshold.

26. (Original) A machine-accessible medium as defined in claim 17, wherein the plurality of machine accessible instructions are structured to cause the device to:

measure the first native software instruction resulting in a first number of bytes;

measure the second native software instruction resulting in a second number of bytes; and

compare the first number of bytes and the second number of bytes, wherein executing the first native software instruction is in response to the first number of bytes being less than

the second number of bytes by at least a predetermined threshold.

27.   (Original) A machine-accessible medium as defined in claim 17, wherein the plurality of machine accessible instructions are structured to cause the device to:

measure the first native software instruction resulting in a first number of bytes;

measure the second native software instruction resulting in a second number of bytes;

and

compare the first number of bytes and the second number of bytes, wherein executing the second native software instruction is in response to the first number of bytes not being less than the second number of bytes by at least a predetermined threshold.

28.   (Original) An apparatus structured to execute a mixed mode code, the apparatus comprising:

a memory device; and

a mixed mode processor operatively coupled to the memory device, the mixed mode processor being structured to execute a runtime environment, the runtime environment being stored in the memory device, the runtime environment comprising:

a compiled binary;

a first code generator to generate a first software instruction based on the compiled binary, the first software instruction being associated with a first instruction set of the mixed mode processor;

a second code generator to generate a second software instruction based on the compiled binary, the second software instruction being associated with a second instruction

- 9 -

set of the mixed mode processor, wherein the first instruction set is different than the second instruction set; and

an executing code including the first instruction and the second instruction.

29. (Original) An apparatus as defined in claim 28, wherein the first instruction set comprises an X-bit wide instruction set, the second instruction set comprises a Y-bit wide instruction set, and X is less than Y.

30. (Original) An apparatus as defined in claim 28, wherein the first instruction set comprises a 16-bit wide instruction set, and the second instruction set comprises a 32-bit wide instruction set.